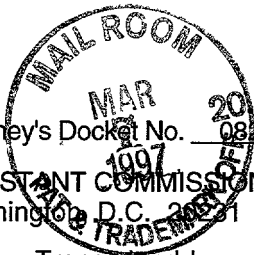


61581 U.S. PTO
08813647
03/07/97



Attorney's Docket No. 082225.P2170

Patent

ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

SIR: Transmitted herewith for filing is the **nonprovisional patent application** of

Inventor(s): Ariel Hendel, Leo Hejza, and Howard Frazier

For: METHOD AND APPARATUS FOR PARALLEL TRUNKING OF INTERFACES TO
INCREASE TRANSFER BANDWIDTH

(Title)

Enclosed are:

- ☒ TEN sheet(s) of Drawings.
☐ An Assignment of the invention to _____
☐ Assignment Cover Sheet Form PTO-1595.
☒ A Declaration and Power of Attorney (_____ signed/ XX unsigned).
☐ A Verified Statement to establish Small Entity Status under 37 C.F.R. §§ 1.9 and 1.27.
☒ Postcard Included

The Filing Fee has been calculated as shown below:

	(Col. 1)		(Col. 2)	
For:	No. Filed		No. Extra	
Basic Fee:				
Total Claims:	37	- 20	*	17
Indep. Claims:	6	- 3	*	3
<input type="checkbox"/>	Multiple Dependent Claim(s) Presented			

* If the difference is less than zero,
enter "0" in Col. 2.

SMALL ENTITY	
Rate	Fee
	\$ 385
x 11	\$
x 40	\$
+ 130	\$
TOTAL	\$

OTHER THAN A SMALL ENTITY	
Rate	Fee
	\$ 770
x 22	\$ 374
x 80	\$ 240
+ 260	\$ 0
TOTAL	\$ 1384

- ☒ A check for \$ 1,384.00 for the filing fee is enclosed.
☐ A check for \$ _____ for recordation of the Assignment is enclosed.

"Express Mail" mailing label number EM389986815US

Date of Deposit March 7, 1997

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Traci Pickering

(Typed or printed name of person mailing paper or fee)

Traci Pickering

(Signature of person mailing paper or fee)

X The Commissioner of Patents and Trademarks is hereby authorized to charge payment of the following fees associated with this communication, or credit any overpayment, to our Deposit Account No. 02-2666. **A duplicate copy of this sheet is enclosed.**

 X Any additional filing fees required under 37 C.F.R. § 1.16.

 X Any patent application processing fees under 37 C.F.R. § 1.17.

 X The Commissioner of Patents and Trademarks is hereby authorized to charge payment of the following fees during the pendency of this application, or credit any overpayment, to our Deposit Account No. 02-2666. **A duplicate copy of this sheet is enclosed.**

 X Any processing fees under 37 C.F.R. § 1.17, including any extension fees.

 X Any filing fees under 37 C.F.R. § 1.16 for presentation of extra claims.

 X Send all correspondence to the undersigned at BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, 12400 Wilshire Boulevard, Seventh Floor, Los Angeles, California 90025, and direct all telephone calls to the undersigned at (408) 720-8598.

Respectfully submitted,

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

Date: March 7, 1997

By Lawrence M. Cho
Lawrence M. Cho

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(408) 720-8598

Reg. No.: 39,942



UNITED STATES PATENT APPLICATION
FOR
METHOD AND APPARATUS FOR
PARALLEL TRUNKING OF INTERFACES
TO INCREASE TRANSFER BANDWIDTH

Inventors:

Ariel Hendel
Leo Hejza
and
Howard Frazier

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025-1026
(408) 720-8598

Attorney Docket No.: 082225.P2170

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EM389986815US

Date of Deposit March 7, 1997

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Traci Pickering
(Typed or printed name of person mailing paper or fee)

Traci Pickering
(Signature of person mailing paper or fee)

2007-03-07 14:54:00

1384 - 101A



METHOD AND APPARATUS FOR PARALLEL TRUNKING OF INTERFACES TO INCREASE TRANSFER BANDWIDTH

FIELD OF THE INVENTION

5 The present invention relates to the field of computer networks. More specifically, the present invention relates to a method and apparatus for parallel trunking of interfaces to increase transfer bandwidth between network devices.

BACKGROUND OF THE INVENTION

Local Area Networks (LANs) following the IEEE 802 Standard Architecture are network environments that interconnect end-nodes using various types of network elements. Each of these network elements is subject to its own set of configuration and topology guidelines. These sets of guidelines and rules are intended to deliver a uniform and well defined data link layer behavior over which upper network layers can operate. Examples of this behavior include the emulation of loop-free broadcast domains where any node may communicate with any other node with no prior signaling required, where packet ordering is preserved between any pair of end-nodes, and where every packet is delivered to its destination no more than once.

Figure 1 illustrates a plurality of end-nodes interconnected by a network. End-nodes 110-113 are typically computers capable of sourcing packets to be delivered over the network 120 to other end-nodes. The end-nodes 110-113 are also capable of sinking packets sent to them by other end-nodes in the network 120. Routers are also considered end-nodes for the purposes of this description. Typical network elements used to build LANs are repeaters (hubs) and bridges. In some cases bridges are designated as LAN switches. The IEEE 802.1d Standard for transparent bridges (LAN switches) and IEEE 802.3 Standard for repeaters and end-nodes provide the topological rules that allow the data link layer behavior described.

Any one of the end-nodes 110-113 may send packets to any other end-node connected to the network 120. Packets are delivered at most once per destination. A sequence of packets sourced at one of the end-nodes 110-113 is seen in the same order at a destination end-node connected to the network 120.

5 Figure 2 illustrates a network implementation. The network 200 includes a plurality of repeaters 210 and 211 and switches 220-222. As a rule, links or interfaces connected to repeaters have the same nominal speed or bit rate and links or interfaces connected to switches can have dissimilar speeds or bit rates. An additional property of the switches 220-222 is traffic isolation. Traffic isolation consists of forwarding
10 packets of data only through the links or interfaces where the destination may reside. This property was deliberately defined to increase the overall network capacity. In order to accomplish traffic isolation, switches 220-222 must know which end-nodes 230-236 are reachable via each link or interface. Bridges or switches complying with IEEE 802.1d automatically learn this information with no assistance from the end-nodes 230-
15 236 or any other external agent, and are called transparent bridges.

 Figure 3 illustrates a transparent bridge. Bridge 310 is connected to end-nodes 311-313. A packet from end-node 311 to end-node 312 is not sent to the link where end-node 313 resides because of the property of traffic isolation. The bridge 310 learns the location of the end-nodes 311-313.

20 Network capacity is central to the function of a network. Traditionally, there have been two approaches to increasing network capacity. The first approach involves partitioning the network. The partitioning approach uses switches, bridges, and routers to replace a network of repeaters. By replacing the network of repeaters with more sophisticated hub devices, the flow of packets in the network is better managed and
25 system performance is increased. The second approach involves providing faster link

alternatives. By implementing links that can support higher bandwidth, the network capacity is increased.

Nevertheless, at any given point in time, the choice of link speeds (10, 100, 1000 Mbps) may not match up very well with the amount of sustained throughput that a particular device can support. When switches and high performance routers are used to interconnect multiple links of a given speed, there is a clear need for the inter-switch or inter-router link to be able to support at least some aggregation of the links. If new hardware is required to utilize a newer, higher speed, increased bandwidth network, the utilization of the newer network may not be as attractive from a cost standpoint.

Furthermore, the cost associated with implementing a newer network with more sophisticated hubs or faster links may also not be attractive from a cost standpoint.

Thus, what is needed is a method and apparatus for increasing data transfer bandwidth between network devices, such as end-nodes and switches.

2025-04-17 10:30:37

SUMMARY

A method for interconnecting a first device and a second device in a network is described. The first device and the second device are connected to a plurality of interfaces. The plurality of interfaces emulate a single high-speed interface.

5 A method for creating a multiple interface connection is disclosed. A first identifier is assigned to a first interface and a second interface at the first device. A path between the first device to the second device is identified with the first identifier.

10 A second method for creating a multi-interface connection is disclosed. A first device is connected to a plurality of interfaces. The plurality of interfaces emulate a single high-speed interface.

15 A network is disclosed. The network includes a first device and a second device. A first interface is coupled to the first device and the second device. A second interface is coupled to the first device and the second device. The first interface and the second interface emulate a single high speed interface. According to an embodiment of the present invention, the first interface and the second interface are assigned an identifier that identifies a path between the first device and the second device.

20 A network device is disclosed. The network device includes a first port that is connected to a first interface. The network also includes a second port that is connected to a second interface. A trunking pseudo driver is coupled to the first port and the second port. The trunking pseudo driver allows the first interface and second interface to emulate a single high-speed device.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

- 5 Figure 1 illustrates a plurality of end-nodes interconnected by a network;
 Figure 2 illustrates a network implementation;
 Figure 3 illustrates a transparent switch;
 Figure 4 illustrates a block diagram of a system which may be programmed to
implement the present invention;
- 10 Figure 5 illustrates a network implementing an embodiment of the present
invention;
 Figures 6a illustrates a first device and a second device connected to a trunk
connection;
 Figure 6b illustrates the first device and the second device as a server and a
15 switch;
 Figure 6c illustrates the first device and the second device as switches;
 Figure 6d illustrates the first device and the second device as servers; and
 Figure 7 illustrates a software embodiment of a server interface according to an
embodiment of the present invention.

08313647.030797
252050 248E780

DETAILED DESCRIPTION

A method and apparatus for high speed data transfer is disclosed. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be
5 apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

COMPUTER SYSTEM OVERVIEW

10 Referring to Figure 4, the computer system upon which an embodiment of the present invention can be implemented is shown as 400. Computer system 400 comprises a bus or other communication device 401 that communicates information, and a processor 402 coupled with bus 401 that processes information. System 400 further comprises a random access memory (RAM) or other dynamic storage device 404
15 (referred to as main memory), coupled to bus 401 that stores information and instructions to be executed by processor 402. Main memory 404 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 402. Computer system 400 also comprises a read only memory (ROM) and/or other static storage device 406 coupled to bus 401 that stores
20 static information and instructions for processor 402. Data storage device 407 is coupled to bus 401 and stores information and instructions. A data storage device 407 such as a magnetic disk or optical disk and its corresponding disk drive can be coupled to computer system 400. Network interface 403 is coupled to bus 401. Network interface 403 operates to connect computer system 400 to a network (not shown).

25 Computer system 400 can also be coupled via bus 401 to a display device 421, such as a cathode ray tube (CRT), for displaying information to a computer user. An

input device 422, including alphanumeric and other keys, is typically coupled to bus 401 for communicating information and command selections to processor 402. Another type of user input device is cursor control 423, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 402 and for controlling cursor movement on display 421. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allows the device to specify positions in a plane.

Alternatively, other input devices such as a stylus or pen can be used to interact with the display. A displayed object on a computer screen can be selected by using a stylus or pen to touch the displayed object. The computer detects the selection by implementing a touch sensitive screen. Similarly, a light pen and a light sensitive screen can be used for selecting a displayed object. Such devices may thus detect selection position and the selection as a single operation instead of the "point and click," as in a system incorporating a mouse or trackball. Stylus and pen based input devices as well as touch and light sensitive screens are well known in the art. Such a system may also lack a keyboard such as 422 wherein all interface is provided via the stylus as a writing instrument (like a pen) and the written text is interpreted using optical character recognition (OCR) techniques.

The present invention is related to the use of computer system 400 to facilitate high speed data transfers via a trunk connection. According to one embodiment, facilitating high speed data transfers via a trunk connection is performed by computer system 400 in response to processor 402 executing sequences of instructions contained in memory 404. Such instructions may be read into memory 404 from another computer-readable medium, such as data storage device 407. Execution of the sequences of instructions contained in memory 404 causes processor 402 to facilitate high speed data transfers via the trunk connection, as will be described hereafter. In

alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software.

5

NETWORK OVERVIEW

The present invention increases the capacity of individual network links or interfaces that do not have repeaters at either end while preserving the guidelines specified by IEEE 802 as perceived by other end-nodes and network elements in the network. The capacity is increased by connecting an arbitrary number of similar links or
10 interfaces in parallel. This approach is useful whenever increasing the raw speed of the existing link is not technically or economically feasible, or when the physical proximity makes parallel links more appealing than changing the link to faster interfaces and media types.

Figure 5 illustrates a network according to an embodiment of the present
15 invention. The network 500 includes a plurality of repeaters 510 and 511 and a plurality of switches 520-522. Trunk 540 connects the switch 521 with the switch 522. Trunk 541 connects switch 522 with end-node 533. Trunk 540 and trunk 541 include a plurality of links or interfaces connected in parallel. Connecting a plurality of links or interfaces in parallel increases the capacity in the path between two devices. The present
20 invention implements trunks while preserving the properties of a IEEE 802 network by preserving a behavior that is transparent to and inter-operable with all other end-nodes and network elements that do not participate in the trunk.

Preserving the properties of IEEE 802 requires addressing the following problems:

25

- 1) A conventional switch connected to the same media access control (MAC) address over more than one link would only use one of these lines (the one learned last);

2) A trunk is a loop, and loops between switches can be broken by 802.1d;

3) Parallel paths may cause packet re-ordering;

4) A conventional 802.1d switch delivers multiple copies of a packet when an end-node is multi-homed;

5 5) A conventional 802.1d switch could loop packets back over the other links of a multi-homed end-node.

Each device connected to the trunks 540 and 541 has a trunking layer. The trunking layer is responsible for load balancing to determine which link or interface to use to transmit a given packet of data. Load balancing is applicable to both end-nodes and switches. The layer has additional duties in the cases of switches, including eliminating looped packets, suppressing multiple copies of a packet, and treating the entire trunk connection as a single logical port in its topology database. The trunking layer uses the same MAC address for all the links or interfaces on a trunk to maximize the transparency towards the protocol stack executing in the end-nodes. By using the same MAC address, a single IP address may be associated with the entire trunk.

Figure 6a illustrates a first device 610 and a second device 620 connected to a trunk 630 that includes a plurality of links or interfaces 631-633. The first device 610 and the second device 620 may be the switch 521 and the switch 522 connected to the trunk 540 or the switch 522 and the end-node 533 connected to the trunk 541 in Figure 5. Alternatively, the first device 610 and the second device 620 may be an end-node, such as a server, client, or router, or a switch as illustrated in Figures 6b-6d. Figure 6b illustrates an embodiment of the present invention where the first device 610 is a server and the second device 620 is a switch. Switch 620 is coupled to a plurality of client segments labeled 641-645. The server 610 may be implemented by the computer system 400 illustrated in Figure 4. Figure 6c illustrates an embodiment of the present invention where the first and second devices 610 and 620 are switches. The switch 610

is coupled to a plurality of client segments labeled 651-655 and the switch 620 is coupled to a plurality of client segments labeled 641-645. Figure 6d illustrates an embodiment of the present invention where the first and second devices 610 and 620 are servers.

5 Figure 7 illustrates a software embodiment of an interface of a computer system connected to a trunk according to an embodiment of the present invention. Upper layers 710 represent any application program that produces or consumes data in the computer system. IP 720 represents an Internet Protocol (IP) layer that makes IP-addressed packets possible. Network device driver 740 represents a layer in the operating system
10 that facilitates communication between hardware and the operating system. Device Units 751-753 represent the physical hardware connected to each of the interfaces 631-633. According to an embodiment of the present invention, interfaces 631-633 are connected to a network device via ports. The trunking pseudo driver 730 resides between the IP layer 720 and the network device driver 740 and contains the trunking
15 layer described above. The trunking pseudo driver 730 splits data in the transmit path and merges data in the receive path of the interfaces 631-633. It should be appreciated that the trunking pseudo driver may be implemented by any known circuitry in a hardware environment.

20 TRUNK CONNECTION ASSIGNED A IDENTIFIER

Referring back to Figure 6a, the plurality of interfaces 631-633 operate to provide a high bandwidth connection between the first device 610 and the second device 620. The physical interfaces 631-633 share a common source device and destination device with each other. The number of interfaces that are implemented may be any
25 number greater than two and dependent on the bandwidth requirement of the network 200; and "trunk" as used herein refers to any such multiple-interface connection, i.e. a

connection having at least two links or interfaces. The plurality of interface 631-633 are assigned an associated identifier that identifies the connection between the first device 610 and the second device 620. For end-nodes, the identifier may be a logical name such as a media access control (MAC) address or an Internet Protocol (IP) address. For
5 switches, the identifier may be a grouping identifier with local significance only. End-nodes connected to the trunk 630 will associate all the interfaces 631-633 of the trunk 630 by its identifier. According to an embodiment of the present invention, interfaces 631-633 are Ethernet interfaces, but may be any suitable network interface such as an Intranet interface (such as LAN) or Internet interface. Interfaces 631-633 may be
10 homogeneous, having identical physical layer and media access control layer characteristics, or non-homogeneous. According to a preferred embodiment of the present invention, the physical and media access control layers for each interface are full duplex.

15 LOAD BALANCING

In order to maximize the throughput rate of data transmitted on the trunk 630, the first device 610 and the second device 620 select one of the interfaces 631-633 in the trunk 630 and uses the selected interface to transmit data. Load balancing in end-nodes typically involves utilizing state information regarding previously sent data, and the
20 status of output queues corresponding to the plurality of interfaces 631-633 in selecting an interface to transmit present data. State information regarding previously sent data is available to the end-node because the software generating the data is running in the same environment as the trunked end-node interface. The depth of the output queue is used as a metric for determining how busy a physical interface is.

25 The temporal ordering of the packets must be preserved when a stream of packets is to be transmitted from one end-node to one or more end-nodes. In order to

satisfy the temporal ordering, the end-node will attempt to ensure that all of the packets associated with a particular transport layer datagram are enqueued on the same network device transmit queue.

According to one embodiment of the present invention, the end-node inspects the header of each packet and uses the information to associated each packet with a particular connection. The end-node keeps a small cache of MAC or IP destination addresses associated with each network interface. When the IP hands the end-node a packet, the end-node checks the cache to see if it has recently transmitted a packet to this destination address. If it has, the end-node will enqueue the packet on the same interface that it enqueued the last packet to this destination address. If the destination address has not been transmitted to recently, the pseudo driver can either enqueue the packet on the last busy transmit queue or the emptiest queue, or the next available queue in a round robin fashion. The end-node updates the cache for that queue with the new destination address. In the situation where the server is sending data to one client, this technique would ensure that all packets to the client travel over the same interface on the trunk.

Load balancing in switches typically involves selecting an interface based on the source address of the packet, or of the packet's port of arrival. The interface selected could, for example, be looked up on a table or calculated using a deterministic algorithm. This scheme results in a static load balancing function that forwards most of the traffic along the same physical interface. As an improvement, it is possible to have a dynamic mapping function and still maintain frame ordering, given that the function changes are slower than the output queue transit times. For instance, the mapping for a given source address can be determined at the time the first packet with the source address is seen, and eventually aged when the source address is not seen for a period of time. By considering both the source address and the port of arrival, the dynamic

00013647-030797
252030743647

mapping function reduces the number of pathological cases. For example, if the traffic is spatially dominated by a particular input port, considering the source address helps spread its traffic, and conversely the port of arrival helps distribute traffic dominated by a small number of addresses in particular if more than one trunk connection exists in the switch.

In an alternate embodiment of the present invention, the mapping function separates traffic according to priority or whether the traffic is bandwidth managed. A priority based mapping function is desirable when packet order preservation is not necessary.

Referring back to Figure 6a, it should be appreciated that the load balancing techniques implemented on device 610 and device 620 do not have to be the same in order to implement the trunk 630. It should also be appreciated that the interfaces 631-633 may be used to transmit data packets bi-directionally.

LOOP PREVENTION

Prevention of frame duplication is achieved at the switch is achieved by treating the set of trunked ports on the switch as if they were a single port with a separate queue per physical interface. Forwarding is performed to only one of its queues. Thus, packets of data with broadcast, group, multicast, or unknown unicast address are not replicated or duplicated across the interfaces 631-633 of the trunk 630. Data transmitted through an interface of the trunk 630 are also not sent back through another interface of the trunk 630.

In the foregoing description, the invention is described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modification and changes may be made thereto without departing from the broader spirit and scope of

the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

Additional details regarding implementations of the present invention are attached in the Appendix.

2025-04-24 14:54:54

Simple Trunking Model (STruM)

Howard Frazier, Leo Hejza, Ariel Hendel
SMCC/NPG
March 3, 1997

1. Introduction

Ethernet has learned a new trick in the past couple of years. It has learned how to scale its link speed by a factor of ten. 10BASE-T begot 100BASE-T, which in turn will be scaled up to one gigabit per second with 1000BASE-T. The Ethernet community has deliberately and emphatically rejected the idea of specifying link speeds in anything other than multiples of 10. Proposals have been made, and shot down in flames, for intermediate link speeds between 10 and 100 Mbps, and they have recently been made again in the context of Gigabit Ethernet, and they have been met with the same reception. No one who sells networks or computers wants to confuse the marketplace by producing multiple, non-interoperable, intermediate link speeds, and everyone seems to feel more comfortable when the link speed can be expressed as power of the number of our fingers or toes.

Never the less, at any given point in time, the choice of link speeds (10, 100, 1000 Mbps) may not match up very well with the amount of sustained throughput that a particular device can support. Virtually all multi-processor servers shipped today can sustain greater than 100 Mbps aggregate network transfer rates. Furthermore, when switches and high performance routers are used to interconnect multiple links of a given speed, there is a clear need for the inter-switch or inter-router link to be able to support at least some aggregation of the links. The next power of ten increase in link speed may not be an attractive choice from a cost standpoint unless the utilization of the higher speed link is going to be greater than 40 to 50 percent. Kicking up the link speed also requires new hardware.

For the purposes of this discussion, trunking will be defined as the ability to combine multiple parallel physical links into one logical channel. We will limit ourselves to trunks in which the physical links share a common source, and a common destination. We will further limit ourselves to trunks in which each of the links (or "segments" of the trunk) have identical physical layer and media access control layer characteristics. We have paid particular attention to the way IP packets will be transported over these trunks, but we don't believe that the model described herein is in any way limited IP networks.

The remainder of this paper will describe a set of rules that the equipment on each end of the trunk must agree to. The rules can be applied to either end stations (DTEs, such as computers of any classification) or network infrastructure components (specifically switches). The rules are symmetric, which is to say that both ends are subject to the same rules.

2. Rules

1. A trunk may have any number of segments, but all segments must have identical physical layer and media access control layer characteristics
2. Each segment of the trunk shares a common source and a common destination with the other segments of the trunk
3. Temporal ordering of the packets transported across a given segment of the trunk must be preserved throughout the network, subject only to loss due to bit errors
4. Temporal ordering of the packets transported across different segments of the trunk must not be assumed
5. Packets must not be replicated or duplicated across the segments of a trunk. This includes broadcast and multicast packets
6. Broadcast and multicast packets transmitted through a segment of the trunk must not be "echoed" or "looped-back" to the sender over the other segments of the trunk
7. The model assumes full duplex operation at the physical and media access control layers for each segment. Half duplex operation, using CSMA/CD, is neither supported nor desired
8. End stations connected to trunks will associate a single 48 bit IEEE MAC address with all segments of the trunk.
9. Load balancing across the segments is not assumed to be perfect. Each end of the trunk will attempt to load balance across the segments to the best of its ability, subject to all of the forgoing rules

These rules do not address the configuration, setup, management, or maintenance of trunks, nor do they address failure detection or recovery. It is assumed that the physical layer will provide some indication if a particular segment of the trunk fails, and that each end of the trunk will monitor whatever status is provided by the physical layer, and take whatever action is deemed appropriate. Configuration and setup are assumed to be performed via manual operations specific to each implementation.

3. Proposals for load balancing

The diagrams in Figure 1 through Figure 3 may assist the reader in understanding the proposals

Figure 1 shows a trunk connection between a server (as a specific example of a DTE) and a switch. The example shows a trunk with 3 segments, labeled A, B, and C. The switch is also connected to several client segments, labeled a, b, c, etc. In this configuration, it is

possible to replace the server with other types of equipment, such as a router, or a high performance workstation, or a printer, to list a few examples.

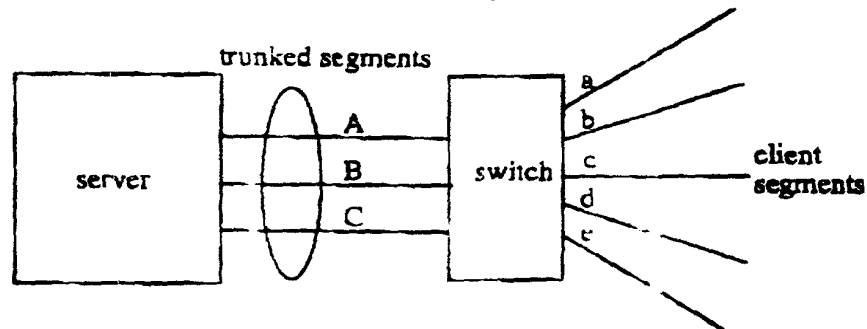


FIGURE 1. Trunks between servers and switches

Figure 2 shows a trunk used as a connection between two switches. As in the previous example, there is no special significance to the number of segments which make up the trunk in Figure 2. The trunk could just as easily be made of two or four or practically any number of segments, depending on the amount of bandwidth required.

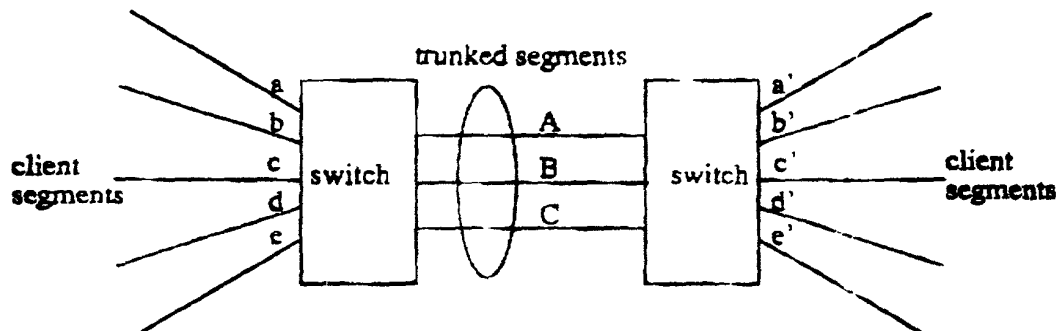


FIGURE 2. Trunks between switches

Figure 3 shows a trunk employed between two servers. Once again, either or both of the servers could be replaced with some other type of equipment, such as a router.

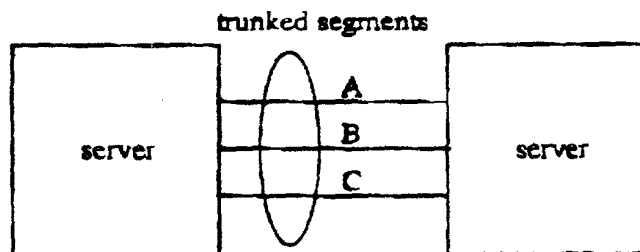


FIGURE 3. Trunks between servers

3.1 Load balancing in servers

An important goal of the model is to make the trunk appear like a single high bandwidth interface from the viewpoint of the server's protocol stack. In addition, all clients which communicate with the server via the trunk should have a consistent view of the server's identity (MAC and IP host addresses). Load balancing by managing the Address Resolution Protocol (ARP) tables in the clients has been considered and rejected because it would dramatically increase the number of ARP frames emitted by the server, and would require a significant amount of added functionality in the server's ARP implementation.

In order to satisfy Rule # 4, "Temporal ordering of the packets transported across different segments of the trunk must not be assumed" the server must ensure that all packets of any sequence of packets which requires temporal ordering are transmitted over the same segment of the trunk.

It should be noted that transport protocols generally can recover from situations where packets arrive out of order, but that this generally entails a significant degradation in throughput, because out of order reception is handled as an exception, and is not optimized. Therefore, the server load balancing mechanism should be designed to take advantage of Rule # 3, "Temporal ordering of the packets transported across a given segment of the trunk must be preserved throughout the network, subject only to loss due to bit errors".

At this point, it might be helpful to introduce a diagram which shows the software components of a trunked server interface.

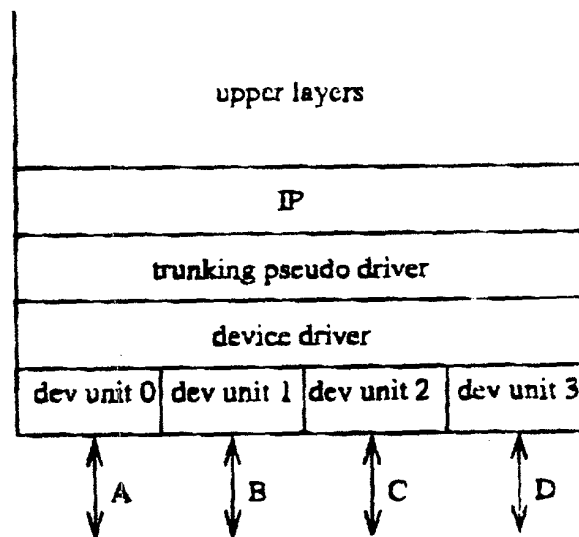


FIGURE 4. Software components of a trunked server interface

A "trunking pseudo driver" is introduced between the IP protocol layer and the network device driver. The function of the pseudo driver is to act as a demultiplexor in the transmit path, and a multiplexor in the receive path. In order to satisfy the rules regarding temporal ordering, the pseudo driver will attempt to ensure that all of the packets associated with a

particular transport layer datagram are enqueued on the same network device transmit queue. This assumption is made on the basis that ordering within a datagram is sufficient, and that ordering between datagrams is unnecessary.

It would be quite difficult for the pseudo driver code to inspect the headers of each packet and attempt to associate them with a particular datagram, though one can imagine that this could be accomplished given an arbitrarily fast processor to execute the code. As a first approximation, the authors feel that it would be sufficient to keep a small cache of MAC (or IP) destination addresses associated with each network interface (each segment of the trunk). Thus, when IP hands the pseudo driver a packet, the pseudo driver checks the cache to see if it has recently transmitted a packet to this DA. If it has, the pseudo driver will enqueue the packet on the same interface that it enqueued the last packet to this DA. If this DA has not been transmitted to recently, the pseudo driver can either enqueue the packet on the least busy transmit queue (the emptiest queue), or the next available queue in a round robin fashion. What ever queue it selects, the driver must update the cache for that queue with the new DA.

In the degenerate case in which a server is talking to one and only one client, this technique would ensure that all packets to that client travel over the same interface, and hence the same segment of the trunk. Why do we call this load balancing? Because it works much better in a non-degenerate case, and will do a good job of ensuring ordered delivery if we can safely assume that the degree of interleaving of packets to different DAs between IP and the network driver is of the same order as the number of processors in a given server. As a first guess, the depth of the cache should be equal to roughly twice the number of processors in a given server. The deeper the cache, the more casual the updating to the cache can be. Experimentation to derive the optimal value for the depth of the cache, and to explore the trade-offs between caching layer 2 and layer 3 addresses is warranted.

3.2 Load balancing in switches

Several switch load balancing mechanisms are possible for forwarding packets into a trunk. The set of load balancing guidelines listed below apply to both switch to switch and switch to server trunks. They ensure that the switch behavior is consistent with conventional bridging guidelines.

1. No frame misordering for a given priority level between a given MAC source and destination.
2. No frame duplication.
3. Transparent to protocols operating above the MAC layer

A natural approach to load balancing is to emulate a faster link by keeping all trunk segments equally busy, possibly by using the corresponding output queue as the metric for how busy the segment is. As long as the links implement flow control, the output queue length is a good end to end proxy for the segment utilization (without flow control, a high segment load is not necessarily reflected in the state of the output queue due to packet loss on the receive queue at the other end).

Deciding for every packet which segment to use, based solely on queue length, might lead to the frame misordering prohibited by the first guideline. If the decision is only a function of the source address of the packet, or of the packet's port of arrival then the first guideline is always satisfied. This scheme however results in a static load balancing function, and the trunk effectiveness depends on the distribution of the traffic sources. While we anticipate that large number of traffic sources would result in acceptably even distributions, it is still possible to end up with configurations where the mapping function forwards most of the traffic to the same segment.

To handle these cases, it is possible to have a dynamic mapping function and still maintain frame ordering, as long as the function changes are slower than the output queue transit times. For instance, the mapping for a given source address can be determined at the time the first packet with the source address is seen, and eventually aged when the source address is not seen for a period of time.

Having the mapping function consider both the source address and the port of arrival reduces the number pathological cases. For example if the traffic is spatially dominated by a particular input port, considering the source address helps spread its traffic, and conversely the port of arrival helps distribute traffic dominated by a small number of addresses (servers or routers) in particular if more than one trunk exists in the switch.

Prevention of frame duplication is achieved by treating the set of trunked ports as if they were a single port, with a separate queue per segment, and making sure that all forwarding is done to only one of its queues.

Furthermore, for the purposes of other 802.1d functions like learning MAC addresses, filtering frames, and executing the Spanning Tree Protocol (if applicable) trunked ports are also treated as if they were a single port.

So far the discussion was centered around using MAC layer information for load balancing. It is possible for the switch to observe higher level protocol information in order to make better load balancing decisions, as long as the third rule of protocol transparency is followed. Transparency implies that the protocols are not aware nor explicitly cooperate with the switch load balancing function. In addition, for protocols that are not supported or understood by the switch, connectivity must be still guaranteed.

Load balancing based on higher level information is practical for switches that examine Layer 3 headers on a packet by packet basis. Many switches examine Layer 3 headers once, for VLAN configuration for example, and use the corresponding Layer 2 information for packet processing. The potential load balancing merits of this approach were not considered.

3.2.1 Other Approaches

The aim of the mapping function could also be different than equally balancing the segments. For example it could separate traffic according to priority, or whether the traffic is bandwidth managed or best effort. A priority based approach is supported by the first

guideline, because packet order preservation is not necessary across different priorities. A priority based approach is straightforward whenever the priority information is well defined at the MAC level (for example if VLAN tags are used).

Restating the main observations, we have shown that the switch behavior is conceptually simple, guided by a set of simple rules along with the particular switch architecture, and can be defined independently of the server load balancing behavior.

00013647-030797
162000-44951880

CLAIMS

What is claimed is:

- 1 1. A method for interconnecting a first device and a second device in a
2 network, comprising the step of:
3 connecting the first device and the second device to a plurality of interfaces; and
4 emulating a single high-speed interface with the plurality of interfaces.
- 1 2. The method of Claim 1, further comprising the step of selecting one of
2 the plurality of interfaces to send a packet of data.
- 1 3. The method of Claim 2, wherein the step of selecting one of the plurality
2 of interfaces to send the packet of data comprises utilizing state information in the first
3 device.
- 1 4. The method of Claim 2, wherein the step of selecting one of the plurality
2 of interfaces to send the packet of data comprises utilizing address information in the
3 packet of data.
- 1 5. The method of Claim 1, further comprising the step of transmitting a first
2 packet of data on only one of the plurality of interfaces.
- 1 6. A method for creating a multi-interface connection that connects a first
2 device and a second device, comprising the steps of:
3 assigning a first identifier to a first interface and a second interface at the first
4 device; and

03031364-030797
/67050"/245E550

identifying a path between the first device to the second device with the first identifier.

7. The method of Claim 6, wherein the step of assigning the first identifier to the first interface and the second interface comprises assigning a media access control (MAC) address.

8. The method of Claim 6, wherein the step of assigning the first identifier to the first interface and the second interface comprises assigning an Internet Protocol (IP) address.

9. The method of Claim 6, wherein the step of assigning the first identifier to the first interface and the second interface comprises assigning a grouping identifier.

10. The method of Claim 6, further comprising the step of allocating data to be transmitted on the first interface and the second interface such that data traffic on the first interface and the second interface is approximately the same.

11. The method of Claim 10, wherein the step of allocating data to be transmitted on the first interface and the second interface, comprises:
checking an output queue of the first interface and an output queue of the second interface;
transmitting the data on the first interface when the output queue of the second interface is fuller than the output queue of the first interface and when previous data sent on the first interface is no longer on the first interface; and

transmitting the data on the second interface when the output queue of the first interface is fuller than the output queue of the second interface and when previous data sent on the second interface is no longer on the second interface.

12. The method of Claim 6, further comprising the step of selecting one of the first interface and the second interface to send a packet of data based on address information in the packet of data.

13. The method of Claim 6, further comprising transmitting a first packet of data on only one of the first interface and the second interface.

14. A method for creating a multi-interface connection, comprising:
connecting a first device to a plurality of interfaces;
emulating a single high-speed interface with the plurality of interfaces.

15. The method of Claim 14, further comprising the step of selecting one of the plurality of interfaces to send a packet of data.

16. The method of Claim 15, wherein the step of selecting one of the plurality of interfaces to send the packet of data comprises utilizing state information in the first device.

17. The method of Claim 15, wherein the step of selecting one of the plurality of interfaces to send the packet of data comprises utilizing address information in the packet of data.

1 18. The method of Claim 14, further comprising the step of transmitting a
2 first packet of data on only one of the plurality of interfaces.

1 19. A network, comprising:
2 a first device;
3 a second device;
4 a first interface coupled to the first device and the second device;
5 a second interface coupled to the first device and the second device, wherein the
6 first interface and the second interface emulate a single high speed interface.

1 20. The network of Claim 19, wherein the first interface and the second
2 interface are homogeneous.

1 21. The network of Claim 19, wherein the first device comprises a load
2 balancing unit that allocates data to be transmitted on the first interface and the second
3 interface such that data traffic on the first interface and the second interface is
4 approximately the same.

1 22. The network of Claim 19, wherein the first device is an end-node.

1 23. The network of Claim 19, wherein the second device is a switch.

1 24. A network, comprising:
2 a first device;
3 a second device;
4 a first interface coupled to the first device and the second device;

5 a second interface coupled to the first device and the second device, wherein the
6 first interface and the second interface are assigned an identifier that identifies a path
7 between the first device and the second device.

1 25. The network of Claim 24, wherein the identifier is an Internet Protocol
2 (IP) address.

1 26. The network of Claim 24, wherein the identifier is a media access control
2 (MAC) address.

1 27. The network of Claim 24, wherein the identifier is a grouping identifier.

1 28. The network of Claim 24, wherein the first interface and the second
2 interface are homogeneous.

1 29. The network of Claim 24, wherein the first device comprises a load
2 balancing unit that allocates data to be transmitted on the first interface and the second
3 interface such that data traffic on the first interface and the second interface is
4 approximately the same.

1 30. The network of Claim 24, wherein the first device is an end-node.

1 31. The network of Claim 24, wherein the second device is a switch.

1 32. A network device, comprising:
2 a first port that connects to a first interface;

3 a second port that connects to a second interface;
4 a trunking pseudo driver, coupled to the first port and the second port, that
5 allows the first interface and second interface to emulate a single high-speed device.

1 33. The network device of Claim 32, wherein the trunking pseudo driver
2 comprises a load balancing unit that selects one of the first and second interfaces to
3 transmit a packet of data.

1 34. The network device of Claim 32, wherein the trunking pseudo driver
2 comprises an identification unit that assigns a first identifier to the first interface and the
3 second interface that identifies a path between the first and the second device.

1 35. The network device of Claim 32, wherein the first and second interface
2 are homogeneous.

1 36. The network device of Claim 32, wherein the network device is an end-
2 node.

1 37. The network device of Claim 32, wherein the network device is a switch.

ABSTRACT OF THE DISCLOSURE

A method and apparatus for interconnecting a first device and a second device in a network. The first device and the second device are connected to a plurality of interfaces. The plurality of interfaces emulate a single high-speed interface. According to an embodiment of the present invention, a first identifier is assigned to the first interface and the second interface at the first device. According to another embodiment of the present invention, one of the plurality of interfaces is selected to transmit a packet of data.

2009-03-04 14:51:34

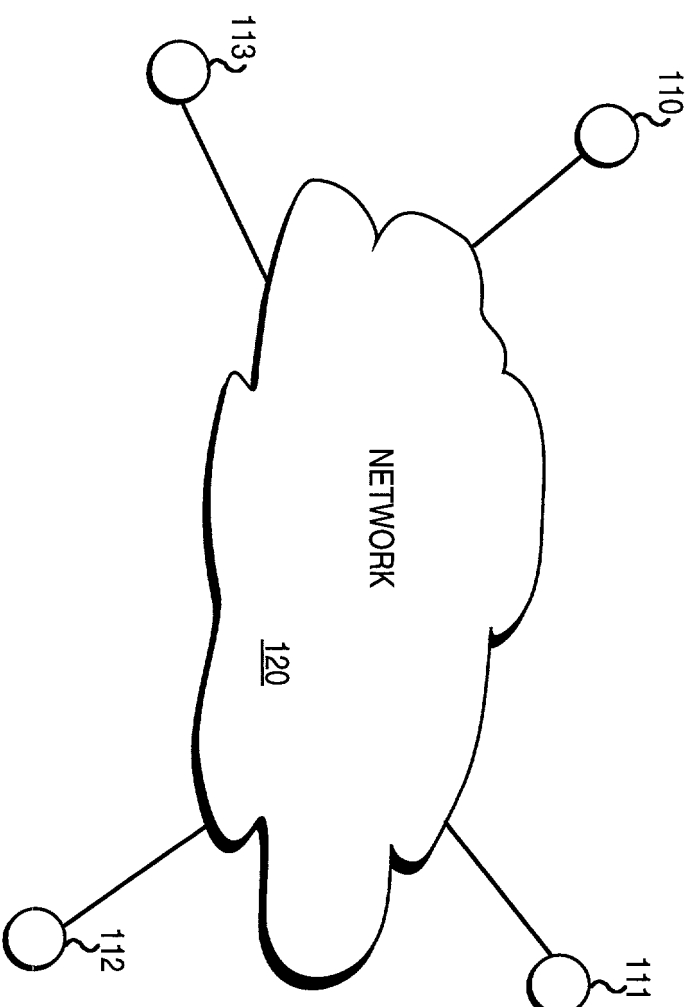


FIGURE 1
(PRIOR ART)

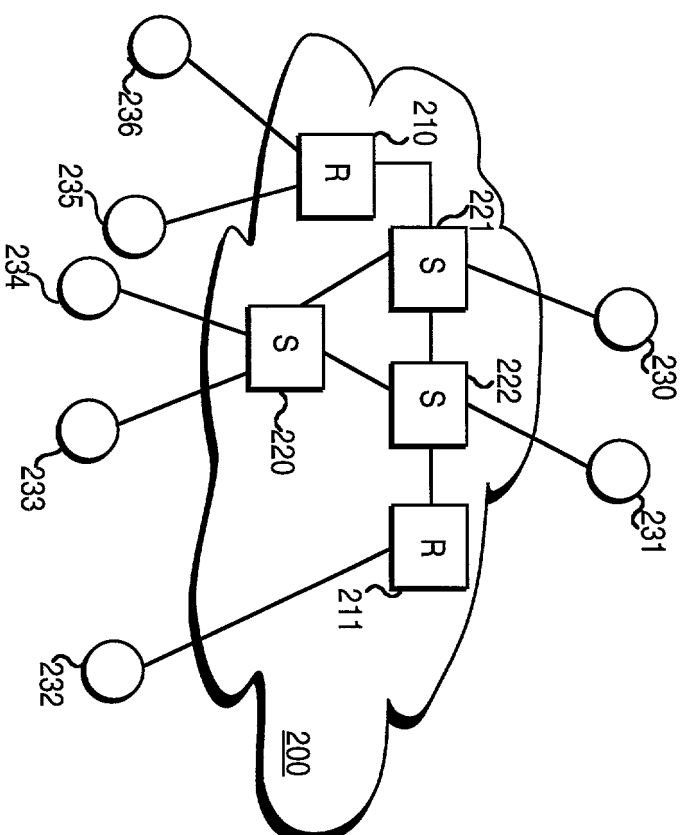


FIGURE 2
(PRIOR ART)

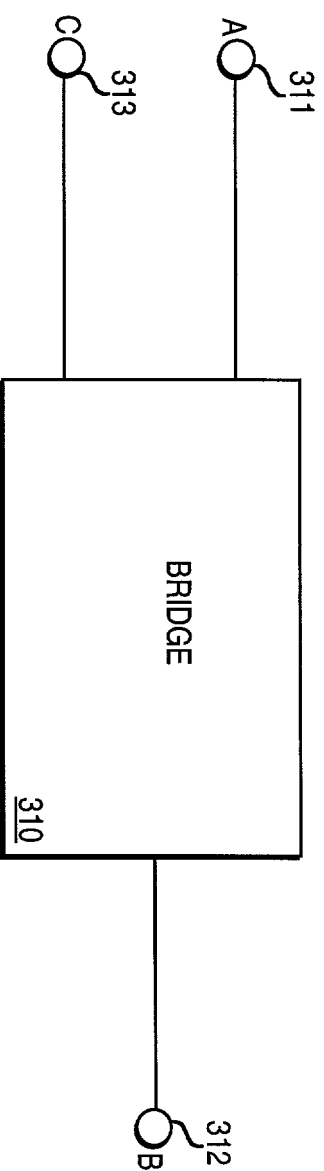


FIGURE 3
(PRIOR ART)

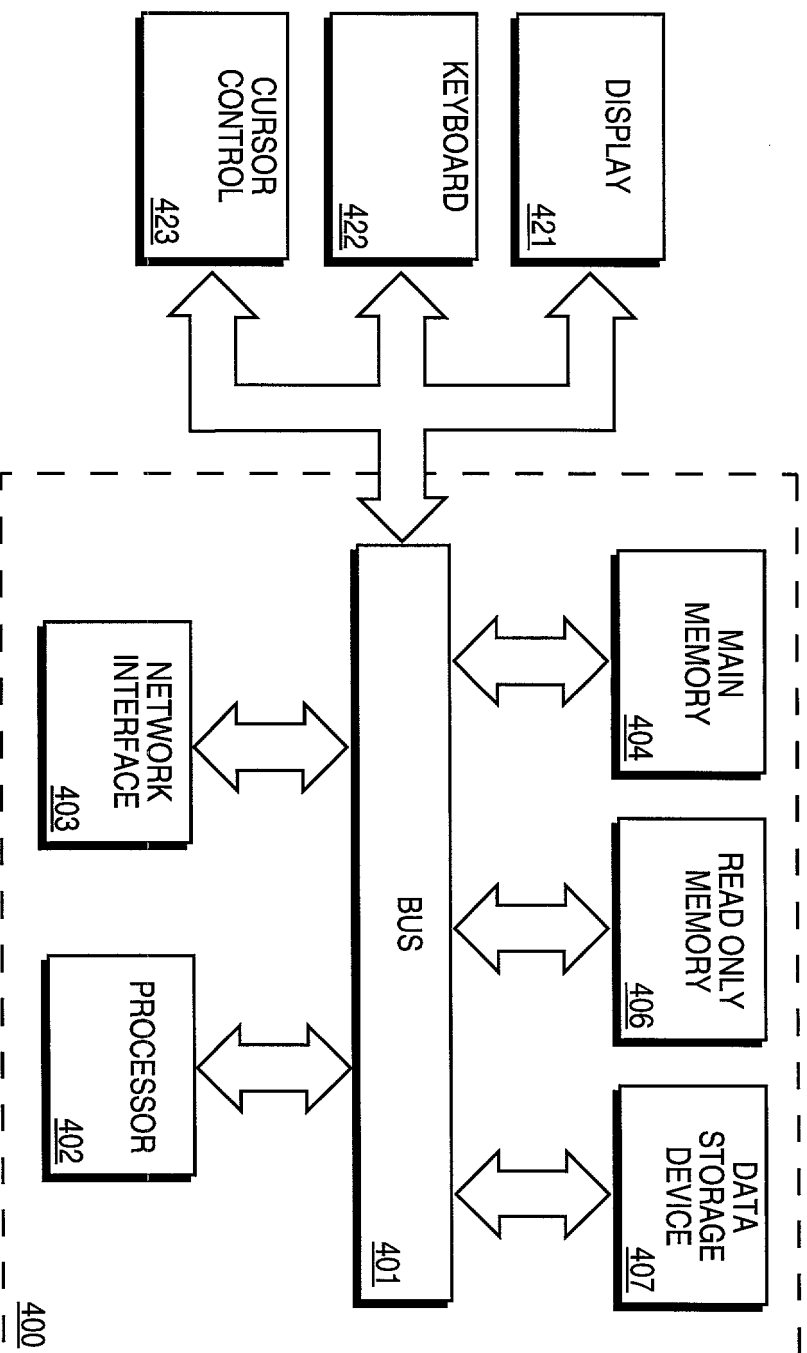


FIGURE 4

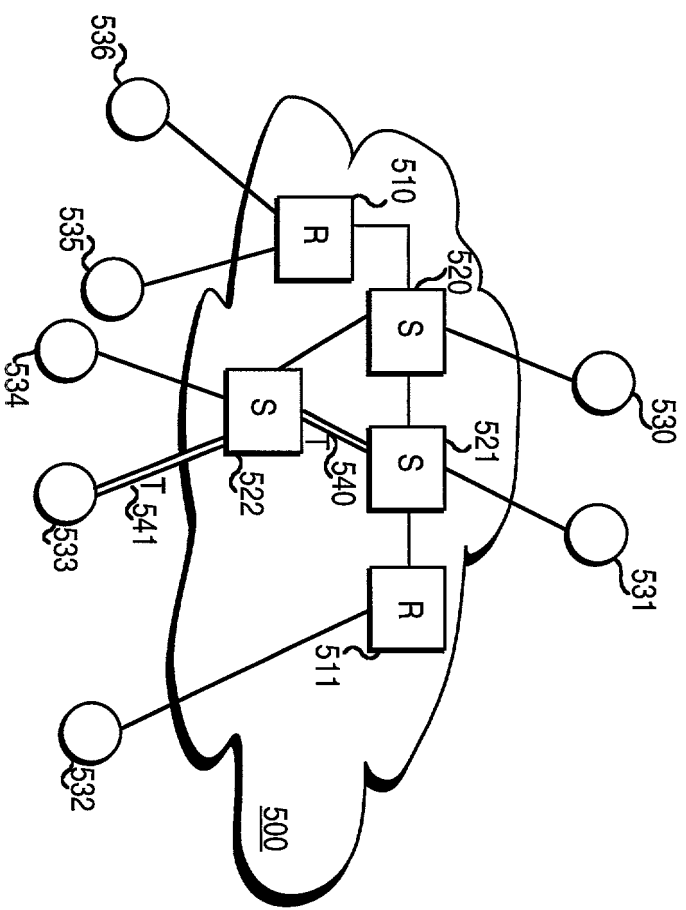


FIGURE 5

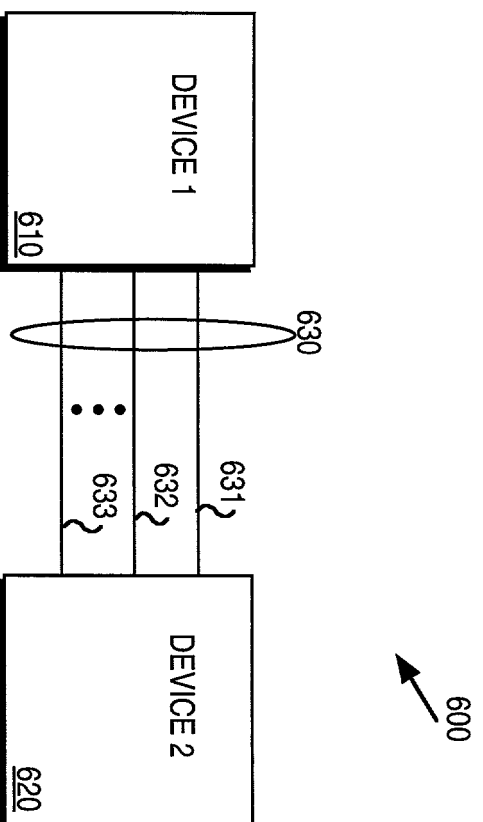


FIGURE 6a

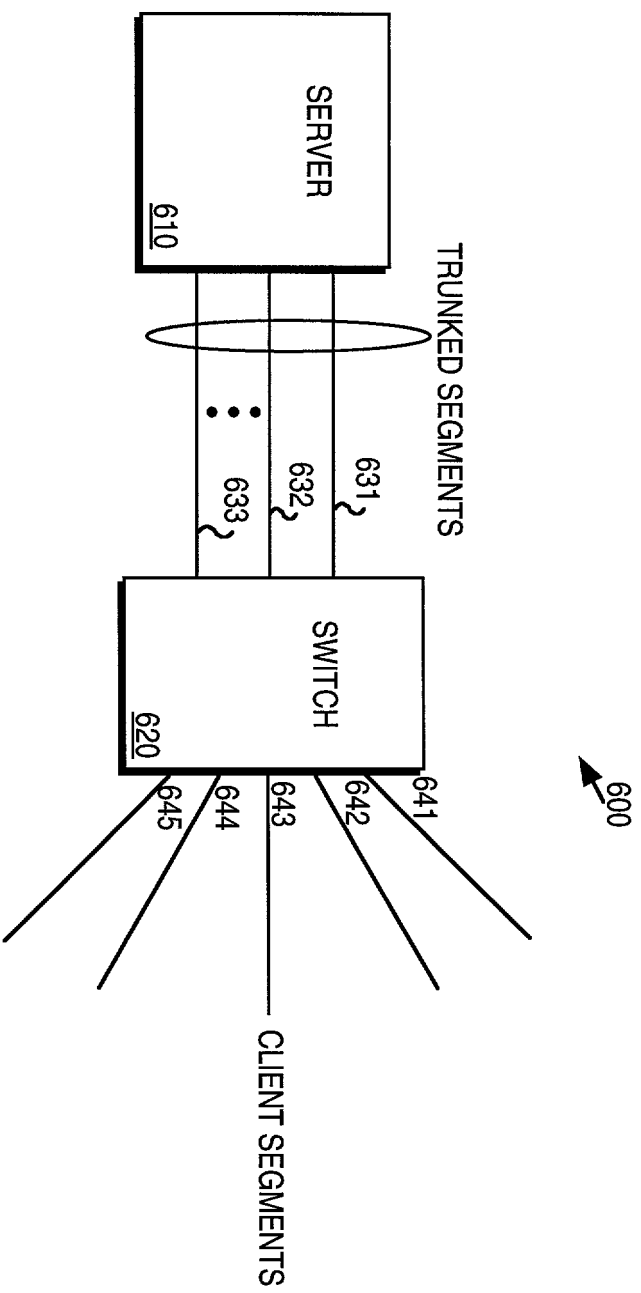


FIGURE 6b

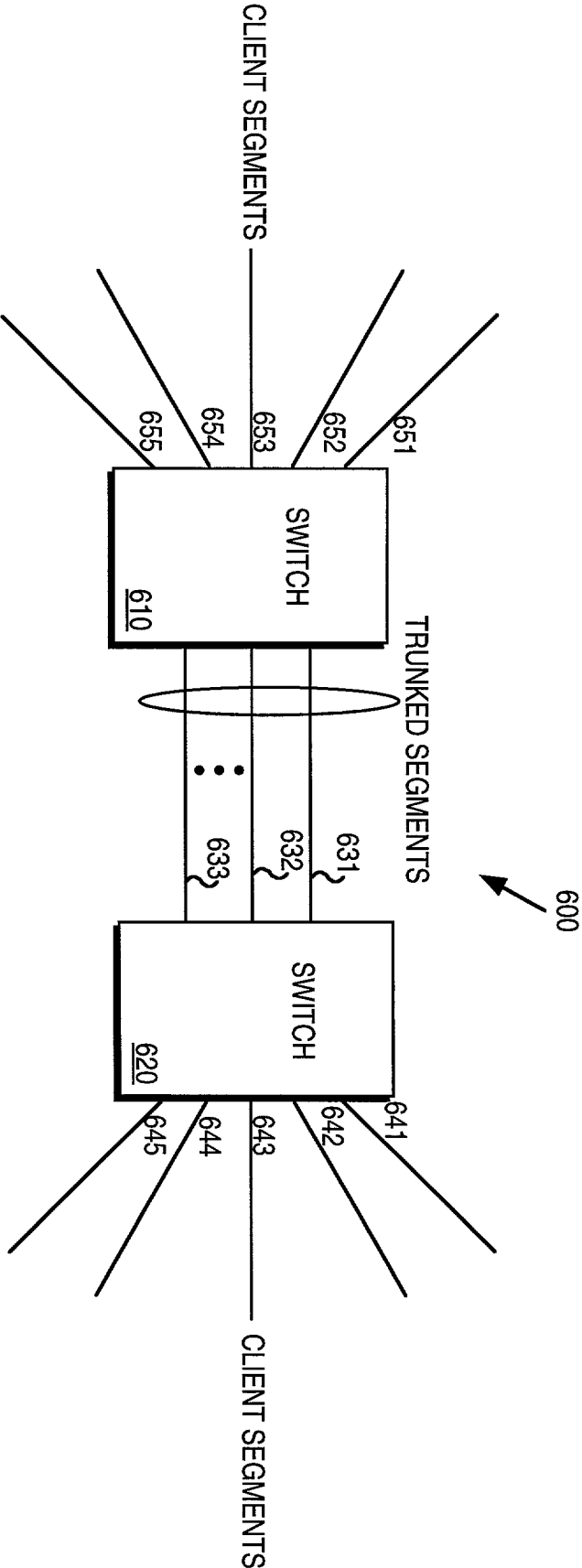


FIGURE 6c

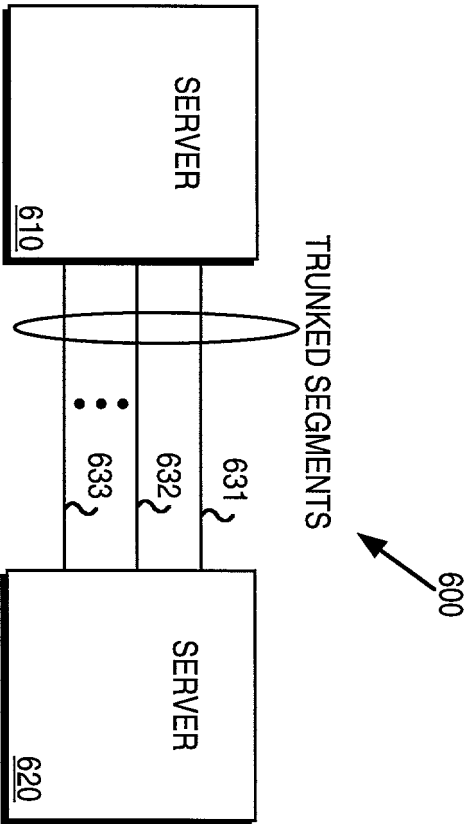


FIGURE 6d

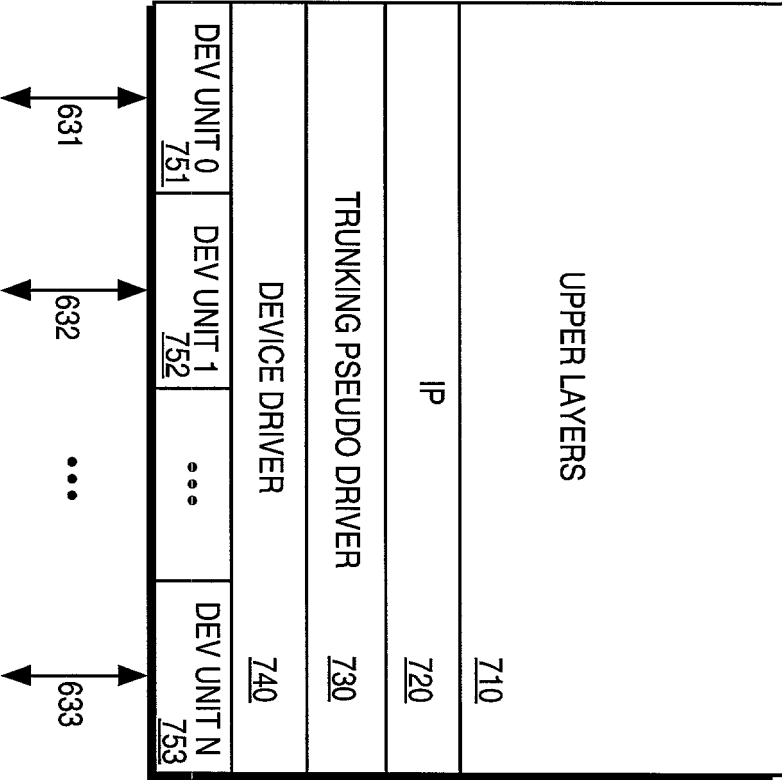


FIGURE 7

Attorney's Docket No.: 082225.P2170

Patent

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND APPARATUS FOR PARALLEL TRUNKING OF INTERFACES TO INCREASE
TRANSFER BANDWIDTH

the specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Rev. 10/01/96 (D1) cak

"Express Mail" mailing label number EM389986815US

Date of Deposit March 7, 1997

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

Traci Pickering

(Typed or printed name of person mailing paper or fee)

Traci Pickering

(Signature of person mailing paper or fee)

Prior Foreign Application(s)

Priority
Claimed

_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	_____ Yes	_____ No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	_____ Yes	_____ No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year Filed)	_____ Yes	_____ No

I hereby claim the benefit under title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below

_____ (Application Number)	_____ Filing Date
_____ (Application Number)	_____ Filing Date

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

_____ (Application Number)	_____ Filing Date	_____ (Status -- patented, pending, abandoned)
_____ (Application Number)	_____ Filing Date	_____ (Status -- patented, pending, abandoned)

00013647-030797
252050-24957880

I hereby appoint Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. 39,591; Jordan Michael Becker, Reg. No. 39,602; Bradley J. Berezna, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; Gregory D. Caldwell, Reg. No. 39,926; Kent M. Chen, Reg. No. 39,630; Lawrence M. Cho, Reg. No. 39,942; Thomas M. Coester, Reg. No. 39,637; Roland B. Cortes, Reg. No. 39,152; William Donald Davis, Reg. No. 38,428; Daniel M. De Vos, Reg. No. 37,813; Karen L. Feisthamel, Reg. No. 40,264; David R. Halvorson, Reg. No. 33,395; Eric Ho, Reg. No. 39,711; George W. Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; Dag H. Johansen, Reg. No. 36,172; Stephen L. King, Reg. No. 19,180; Dolly M. Lee, Reg. No. 39,742; Michael J. Mallie, Reg. No. 36,591; Kimberley G. Nobles, Reg. No. 38,255; Ronald W. Reagin, Reg. No. 20,340; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Allan T. Sponseller, Reg. No. 38,318; Steven R. Sponseller, Reg. No. 39,384; David R. Stevens, Reg. No. 38,626; Edwin H. Taylor, Reg. No. 25,129; Lester J. Vincent, Reg. No. 31,460; John Patrick Ward, Reg. No. 40,216; Ben J. Yorks, Reg. No. 33,609; and Norman Zafman, Reg. No. 26,250; my attorneys; and Gary B. Goates, Reg. No. 35,159; Michael Anthony DeSanctis, Reg. No. 39,957; Charles E. Shemwell, Reg. No. 40,171; Edwin A. Sloane, Reg. No. 34,728; and Judith A. Szepesi, Reg. No. 39,393; my patent agents, of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (310) 207-3800, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Ariel Hendel

Inventor's Signature _____ Date _____

Residence _____ Citizenship _____
(City, State) (Country)

Post Office Address _____

Full Name of Second/Joint Inventor Leo Hejza

Inventor's Signature _____ Date _____

Residence _____ Citizenship _____
(City, State) (Country)

Post Office Address _____

Full Name of Third/Joint Inventor Howard Frazier

Inventor's Signature _____ Date _____

Residence _____ Citizenship _____
(City, State) (Country)

Post Office Address _____

Full Name of Fourth/Joint Inventor _____

Inventor's Signature _____ Date _____

Residence _____ Citizenship _____
(City, State) (Country)

Post Office Address _____

Title 37, Code of Federal Regulations, Section 1.56
Duty to Disclose Information Material to Patentability

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made or record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

- (1) Each inventor named in the application;
- (2) Each attorney or agent who prepares or prosecutes the application; and
- (3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.